Tecnologie Web T (9 cfu)
Prova d'Esame – 26 Gennaio 2022 – Versione A

Tempo a disposizione: 180 minuti

La soluzione comprende la consegna elettronica dei seguenti file:

A1.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
A2.zip	file zip contenente il sorgente java/class e pagine Web per punto 2
A3.zip	file zip contenente pagine Web e codice React.js per punto 3

Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, descrittori, risorse statiche o dinamiche, codice Java e relativi .class, ecc.) e NON dell'intero progetto.

N.B. Per superare la prova scritta di laboratorio ed essere ammessi all'orale, è necessario totalizzare almeno 18 punti (su un totale disponibile di 33), ben distribuiti sui 3 esercizi, ovvero in ciascuno dei tre esercizi si deve raggiungere una valutazione almeno quasi sufficiente.

ESERCIZIO 1 (10 punti)

Si realizzi una applicazione Web per **l'elaborazione casuale server-side di un testo**, basandosi principalmente sulle tecnologie Java Servlet e JSP.

L'applicazione Web deve permettere a utenti autenticati di inserire una stringa *nomeFile* che rappresenti il nome di un file già disponibile server-side; si controlli che tale stringa contenga solo caratteri alfabetici e numerici. All'inserimento del carattere speciale "spazio", il nome file deve essere inviato automaticamente al servitore, senza alcun intervento esplicito dell'utente, per una elaborazione in pipeline del seguente tipo: una **servlet S1** dovrà scegliere casualmente un carattere alfabetico ed eliminarne tutte le occorrenze dal testo di *nomeFile*; il risultato dell'elaborazione di S1 dovrà essere inviato a una **JSP J2**, che dovrà avere il medesimo comportamento e in più calcolare il numero di maiuscole presenti nel testo risultante. Il testo modificato più il conteggio dovranno essere restituiti al cliente in formato JSON.

In ogni momento di esecuzione dell'applicazione, previa autenticazione, un amministratore deve avere la possibilità di visualizzare:

- La lista delle sessioni correntemente attive;
- Il numero di richieste di elaborazione sottomesse in ogni sessione;
- Il numero di richieste totali sottomesse negli ultimi 60 minuti.

Tecnologie Web T (9 cfu)
Prova d'Esame – 26 Gennaio 2022 – Versione A

ESERCIZIO 2 (12 punti)

Si realizzi una applicazione Web per **determinare se una matrice 3 * 3 rappresenta un quadrato magico**. Si definisce "quadrato magico" una qualunque matrice quadrata di numeri naturali in cui sono rispettate le seguenti proprietà: non ci possono essere elementi ripetuti; la somma di ogni riga è uguale alla somma di ogni colonna ed è uguale alla somma di ciascuna delle due diagonali. L'applicazione Web deve essere basata principalmente su tecnologie Javascript e WebSocket.

In particolare, l'applicazione Web deve permettere a ogni utente collegato di potere inserire uno qualunque dei 9 elementi della matrice condivisa considerata, controllando che siano effettivamente numeri naturali, non siano già stati inseriti da altri e non siano ripetuti (tramite controlli puramente locali). Dopo avere terminato l'inserimento di un elemento valido, l'elemento deve essere automaticamente inviato al servitore, senza pressione esplicita di pulsanti, e il servitore deve aggiornare tutti gli utenti collegati tramite WebSocket.
Una volta completato l'inserimento di tutti i 9 elementi, sempre in modo automatico, l'applicazione server-side deve verificare se si tratta di quadrato magico in modo concorrente: una **servlet S1** dovrà controllare che le somme di ogni riga siano uguali; una **servlet S2** dovrà controllare che le somme di ogni colonna e delle due diagonali siano uguali. In particolare, ogni servlet dovrà restituire in modo autonomo e **in formato JSON** un valore "vero"/"falso" a seconda del risultato di tale controllo e, nel caso di "vero", anche il risultato della somma.
Solo dopo avere ricevuto tutti e due i risultati, ogni cliente dovrà visualizzare il risultato complessivo, ovvero se gli elementi inseriti appartengono a un quadrato magico oppure no.

ESERCIZIO 3 (11 punti)

Si realizzi in React.js un'applicazione Web che simuli il gioco "Pesca al Lago". L'applicazione dovrà eseguire interamente sul browser senza interagire con alcun server remoto.

L'interfaccia dell'applicazione sarà composta dalle seguenti sezioni:

- **Sezione Configurazione**. In questa sezione sono presenti due elementi di input per l'inserimento delle dimensioni del lago (rispettivamente, larghezza e lunghezza) e un elemento di input per il numero di lanci che possono essere effettuati dal giocatore in una battuta di pesca. I campi larghezza e lunghezza non possono assumere valori inferiori a 8. Acquisiti tali dati, l'utente potrà iniziare a giocare.
- **Sezione Lago**. Tale sezione deve contenere una griglia rettangolare (larghezza * lunghezza) di celle di colore grigio. Ogni cella nasconde un numero casuale di pesci compreso tra 0 e 5. Il giocatore effettua un lancio cliccando su una determinata cella; effettuato il lancio, sulla cella in questione (cella target) e su tutte le celle immediatamente adiacenti appaiono, rimanendo visibili per tutto il resto della battuta, i numeri corrispondenti di pesci contenuti (e quindi pescati), la cui somma costituirà il punteggio da attribuire al lancio. Tutte le celle interessate da un lancio (cella target e celle adiacenti) non potranno più essere obiettivo di ulteriori lanci; pertanto, il giocatore potrà effettuare i successivi lanci solo su celle in cui non ha ancora pescato. Il gioco termina quando il giocatore avrà esaurito il numero di lanci a disposizione.
- **Sezione Punteggio**. In questa sezione, occorre visualizzare una lista aggiornata in tempo reale con i punteggi ottenuti dal giocatore per ciascun lancio e, alla fine della battuta di pesca, il punteggio complessivo e il punteggio medio per lancio.